University of Tennessee, Knoxville
TRACE: Tennessee Research and Creative Exchange

Chancellor's Honors Program Projects

Supervised Undergraduate Student Research and Creative Work

Spring 5-2006

Load Balancing for High Availability

Matthew Graham Lopez
University of Tennessee-Knoxville

Follow this and additional works at: https://trace.tennessee.edu/utk_chanhonoproj

Recommended Citation

Lopez, Matthew Graham, "Load Balancing for High Availability" (2006). *Chancellor's Honors Program Projects.*
https://trace.tennessee.edu/utk_chanhonoproj/988

This is brought to you for free and open access by the Supervised Undergraduate Student Research and Creative Work at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Chancellor's Honors Program Projects by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

Graham Lopez / Advisor - Dr. Straight

Computer Science - Spring '06

Honors Program Senior Project

Load Balancing for High Availability

Load balancing in the realm of computer science consists of taking a service and improving its availability by using multiple resources to process incoming requests. This technique is used mainly for two different types of service improvement: (1) higher performance, and (2) higher availability or increased reliability. These differing paradigms can be used concurrently or individually. Whereas load balancing for higher performance would be important in applications where an increased throughput is desired such as high performance cluster computing or a high traffic e-service, load balancing for high availability is used for mission-critical services or services where a very high percentage uptime is required.

For my project, I wanted to offer the benefits of high availability load balancing to some of the critical services in the Department of Computer Science. This project would need to be inexpensive, and it would need to be maintained by other individuals after I leave. The major aspects of this project included researching the different types of solutions available, picking a best fit, installation and configuration, and putting resources in place to ease the adaption of the existing services to the high-availability infrastructure.

High availability is achieved through load balancing mainly by eliminating single

points of failure. A single point of failure can consist of any entity on whose presence the functionality of the services depends. Examples can be physical hardware components, software components, or anything else that is necessary to a service being provided. A load balancing solution eliminates single points of failure by monitoring the critical components of the system, and when it detects a failure, automatically failing over to a healthy standby system. This failover is done without needing any intervention by the administer whatsoever, and it is completely transparent to the end users. As a result, if a server providing the load balanced service, or a load balancer itself experiences a failure at any time, any users will not notice, and the problem can be taken care of in the future by the administrator, also without any interruption in service.

Another less obvious advantage of using load balancing for high availability is ease of administration and maintenance to the servers which provide the load balanced services. When the load balancing is implemented correctly, the failover of any particular service is completely transparent to the client. Not only is this useful for unexpected failures, but also for administered failures as well. If hardware or software needs to be replaced on a machine providing a load balanced service, that machine can be taken out of the infrastructure, replaced or repaired, and inserted back into the live pool of load balanced servers. The administrator can carry out all of these maintenance related tasks without causing any inconvenience to the clients.

Since the services which LDAP provides are critical to most other major services in the department which include authentication, web services, and email, I decdided that LDAP was a perfect candidate for the high availability functionality. When this

service becomes unavailable for any reason, most other functions of the department's computing resources are crippled to at least some extent. The aim is to prevent these outages that render authentication, web, email, and all of the desktop systems physically in the department useless. It would also be advantageous to implement the load balancing facility in such a way that these other critical services could themselves be integrated to take advantage of the high availability benefits.

I picked the KeepAlived health checking software from www.keepalived.org to run on the servers that are responsible for the high availability functions of the LDAP infrastructure. I chose this for its stability and reputed reliability. This project, unlike some others, has been around for long enough to have had a chance to mature. Also, being open source software, it was better than some other vendor-offered solutions as far as price was concerned. One of the main developers on the project also released a Debian linux package for it, which implies that the software is well supported on the linux infrastructure which is already in place on the department servers. All of these factors contributed to my choice in software for the load balancing directors.

The load balancing infrastructure is completely scalable to other services offered by the department as well. Since the LDAP service has been running successfully with the load balancing solution in place for several months, we have decided to migrate email, web, and database services to it as well. This will reduce the downtime for those services and make maintenance on these crucial systems easier. In addition to discussing this plan with my supervisors, I have collaborated with the members of the system administration group chiefly in charge of email and web to discuss load

balancing for these services. I have given several presentations on the concept, implementation, and configuration of the load balancing system, and explained how these other parts of our department resources could be incorporated. All parties are interested and plans are in place to add these services to the load balancing infrastructure which I set up for my project. The end result has been so far and will hopefully continue to be the faculty and students of the Computer Science department realizing less downtime of the services on which they depend for their academic and research-related computing needs.